REMARKS

Claims 1-24 were presented for examination. Claims 1-24 have been rejected under 35 U.S.C. § 102(b). Claims 1, 2, 8, 17 and 18 have been amended to clarify the claimed inventions. No new matter has been added. The comments below address all of the stated grounds for rejection and place the pending claims 1-24 in condition for allowance.

I. Claim Amendments

Claim 1 has been amended to include the step of "determining the number of valid instructions based on the edge detection of the valid instructions." This amendment has been made to clarify that the edge detection is used to determine the number of valid instructions.

Claims 2, 8, 17 and 18 have been amended to clarify the edge detecting technique provided by the claimed invention. The edge detecting technique is used to detect a transition from a valid instruction to an invalid instruction. Claim 2 has also been amended to add the step of "determining the number of valid instructions before the transition from the valid instruction to the invalid instruction."

II. Rejection of Claim 1 under 35 U.S.C. § 102(b)

Claim 1 has been rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,098,165 to Penwar et al. (Penwar). Applicants respectfully traverse this rejection and contend that Penwar does not anticipate claim 1, as amended.

The claimed invention allows for processing valid, invalid, and complex instructions. Complex instructions may be broken down into valid and invalid instructions. Claim 1, as amended, is directed to a method for calculating the number of valid instructions within a microprocessor. The method provides that instructions are advance within a microprocessor pipeline and that a transition from a valid instruction to an invalid instruction is detected by within the microprocessor pipeline using an edge detection technique. Based on the edge detection of valid instructions, the number of valid instruction is determined.

Penwar discusses a method of processing complex and non-complex instructions without penalizing the processing of non-complex instructions. Penwar marks complex instructions in a main bundle with a marker bit, and a controller is used to scan the marked instructions in the main bundle, and break up the main bundle of instructions into a smaller bundle that contains only complex instructions. The smaller bundle of complex instructions is passed to helper logic, while the non-complex instructions bypass the helper logic. Therefore, only the complex instructions are passed through the helper logic. It is therefore observed that Penwar allows detection of complex instructions based on the marker bit that the process in Penwar sets and that based on this detection Penwar determines which pipeline the instruction will follow.

Penwar fails to disclose the step of "determining the number of valid instructions based on the edge detection of the valid instructions," as required by amended claim 1. Rather, Penwar provides a process for separating non-complex and complex instructions. Penwar detects complex instructions and creates a smaller bundle that contains only complex instructions. Penwar does not disclose using edge detection to determine the number of valid instructions. The claimed invention is distinguished from Penwar, because the claimed invention uses edge detecting of valid instructions to determine the number of valid instructions.

Further, Penwar fails to disclose the step of "edge detecting a transition from a valid instruction to an invalid instruction," as required by claim 1. As discussed above, Penwar detects complex instructions and separates the complex instructions into a smaller bundle. Penwar, however, does not disclose edge detection as provide by claim 1. That is, the claimed invention provides detecting a transition from a valid instruction to an invalid instruction. The edge detection of the claimed invention is not used for detecting a clock edge, but rather is used to determine a transition from a valid instruction to an invalid instruction. The claimed invention is unlike Penwar, because the claimed invention detects a transition from a valid instruction to an invalid instruction.

For at least this reasons, Applicants respectfully contend that Penwar does not anticipate claim 1. Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claim 1 under 35 U.S.C. § 120(b).

III. Rejection of Claim 2 under 35 U.S.C. § 102(b)

Claim 2 has been rejected under 35 U.S.C. § 102(b) as being anticipated by Penwar. Applicants respectfully traverse this rejection and contend that Penwar does not anticipate claim 2, as amended.

The claimed invention allows for processing valid, invalid, and complex instructions. Complex instructions may be broken down into valid and invalid instructions. Claim 2, as amended, is directed to a method for calculating the number of valid instructions within a microprocessor. The method provides that a bundle of instructions is fetched and that an edge detecting technique is used to detect a transition from a valid instruction to an invalid instruction in the bundle of instructions. The method further determines the number of valid instructions before the transition from the valid instruction to the invalid instruction.

As discussed above, Penwar marks complex instructions in a main bundle with a marker bit, and a controller is used to scan the marked instructions in the main bundle, and break up the main bundle of instructions into a smaller bundle that contains only complex instructions. Penwar allows detection of complex instructions based on the marker bit that the process in Penwar sets and that based on this detection Penwar determines which pipeline the instruction will follow.

Penwar fails to disclose the step of "edge detecting a transition from a valid instruction to an invalid instruction in the bundle of instructions," as required by claim 2. As discussed above, Penwar detects complex instructions and separates the complex instructions into a smaller bundle. Penwar, however, does not disclose edge detection as provide by claim 2. That is, the claimed invention provides detecting a transition from a valid instruction to an invalid instruction. The edge detection of the claimed invention is not used for dctecting a clock edge, but rather is used to determine a transition from a valid instruction to an invalid instruction. The claimed invention is unlike Penwar, because the claimed invention detects a transition from a valid instruction to an invalid instruction.

Notwithstanding the above, Penwar fails to disclose the step of "determining the number of valid instructions before the transition from the valid instruction to the invalid instruction," as

required by claim 2. Penwar simply breaks instructions up based on whether the instruction is complex or non-complex and does not determine the number of valid instructions. Further, Penwar does not disclose determining the number of valid instructions that occur before the transition from a valid instruction to an invalid instruction. The claimed invention is distinguished from Penwar, because the claimed invention requires determining the number of valid instructions before the transition from a valid instruction to an invalid instruction.

For at least this reasons, Applicants respectfully contend that Penwar does not anticipate claim 2. Claims 3-7 depend, directly or indirectly, on claim 2, and therefore incorporate all of the patentable features of claim 2. Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claims 2-7 under 35 U.S.C. § 120(b).

IV. Rejection of Claim 8 under 35 U.S.C. § 102(b)

Claim 8 has been rejected under 35 U.S.C. as being anticipated by Penwar. Applicants respectfully traverse this rejection.

The claimed invention allows for processing valid, invalid, and complex instructions. Complex instructions may be broken down into valid and invalid instructions. Claim 8 provides that a bundle of instructions having a complex instruction is fetched and that at least one instruction occurring after the complex instruction is shifted. Claim 8 further provides determining the number of valid instructions by edge detecting the number of valid instructions occurring after the complex instruction and up to a detection of a transition from a valid instruction to an invalid instruction.

Penwar fails to disclose the step of "determining the number of valid instructions by edge detecting the number of valid instructions occurring after the complex instruction and up to a detection of a transition from a valid instruction to an invalid instruction," as required by claim 8. As discussed above, Penwar detects complex instructions and separates the complex instructions into a smaller bundle. Penwar, however, does not disclose detecting a transition from a valid instruction to an invalid instruction. Nor does Penwar disclose determining the number of valid instructions occurring after the complex instruction and up to a transition from a valid instruction to an invalid instruction. The claimed invention is unlike Penwar, because the

claimed invention detects a transition from a valid instruction to an invalid instruction after a complex instruction and determines the number of valid instructions based on the transition.

For at least this reasons, Applicants respectfully contend that Penwar does not anticipate claim 8. Claims 9-16 depend, directly or indirectly, on claim 8, and therefore incorporate all of the patentable features of claim 8. Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claims 8-16 under 35 U.S.C. § 120(b).

V. Rejection of Claim 17 under 35 U.S.C. § 102(b)

Claim 17 has been rejected under 35 U.S.C. as being anticipated by Penwar. Applicants respectfully traverse this rejection.

The claimed invention allows for processing valid, invalid, and complex instructions. Complex instructions may be broken down into valid and invalid instructions. Claim 17 is directed to fetching a bundle of instructions. The bundle of instructions includes a complex instruction. Claim 17 provides that valid instructions that occur prior to the complex instruction are executed during a first clock cycle, and that the complex instruction is executed during a second clock cycle. Claim 17 further provides the instructions within the bundle occurring after the complex instruction are shifted during at least one of the first clock cycle and the second clock cycle, and edge detecting a transition from a valid instruction to an invalid instruction to determine the number of valid instructions occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle. Claim 17 further provides that the valid instructions occurring after the complex instruction are executed during a third clock cycle.

Penwar discusses parsing a main bundle into sub bundles. In figure 5 of Penwar, a main bundle is parsed into three sub-bundles. The first sub-bundle 500A includes instructions prior to a complex instruction. The sub-second bundle 500B includes the complex instruction and the third sub-bundle 500C includes instructions after the complex instruction. Penwar discloses the sub-bundles represent the sequential contents of the main bundle and that the sub-bundles do not exist at the same time. *Penwar col. 9, line 54-55.* Penwar further provides that the first sub-bundle is created in a first cycle, the second sub-bundle is created in a next or second cycle, and the third sub-bundle is created in a third cycle. Instructions in the third sub-bundle, which

contains non-complex instructions occurring after the complex instruction, are not shifted within the bundle, but rather, the instructions in the third sub-bundle are passed downstream for processing in order. *Penwar col. 10, lines 4-22.*

Penwar fails to disclose the step of "shifting instructions within the bundle occurring after the complex instruction during at least one of the first clock cycle and the second clock cycle." Rather, Penwar provides breaking a main bundle of instruction into sub-bundles and processing the instructions in the sub-bundles in the same order that they reside in the sub-bundles. More specifically, Penwar does not disclose that the instructions in third sub-bundle are shifted. Creating sub-bundles of instructions, where the instructions in the sub-bundles are not shifted is not the same as shifting instructions within a bundle. Even if the examiner construes the creation of sub-bundles in Penwar as shifting, the third sub-bundle is not created until the third clock cycle. The claimed invention is distinguished from Penwar, because the claimed invention requires shifting instructions within a bundle of instructions that occur at the complex instruction in at least the first cycle or the second cycle.

For at least these reasons, Applicants respectfully contend that Penwar does not anticipate claim 17. Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claim 17 under 35 U.S.C. § 120(b).

VI. Rejection of Claim 18 under 35 U.S.C. § 102(b)

Claim 18 has been rejected under 35 U.S.C. § 102(b) as being anticipated by Penwar. Applicants traverse this rejection.

Claim 18, as amended, provides an apparatus for calculating the number of valid instructions within a microprocessor. The apparatus includes a mechanism for fetching a bundle of instructions, and an advancing mechanism for advancing instructions along a microprocessor pipeline. The apparatus also includes an edge detection element for detecting a transition from a valid instruction to an invalid instruction within the bundle.

Penwar fails to disclose the edge detection element of claim 18 that is used for detecting a transition from a valid instruction to an invalid instruction within a bundle. Penwar provides a

process for separating non-complex and complex instructions. Penwar detects complex instructions and creates a smaller bundle that contains only complex instructions. Penwar does not disclose using an edge detection element for detecting a transitions from a valid instruction to an invalid instruction within a bundle. Thus, Penwar fails to disclose all of the patentable features of claim 18.

For at least these reasons, Applicants respectfully contend that Penwar does not anticipate claim 18. Claims 19-24 depend, directly or indirectly, on claim 18, and therefore incorporate all of the patentable features of claim 18. Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claims 18-24 under 35 U.S.C. § 120(b).

Application No.: 10/010389 Docket No.: SMQ-143/P6594

VII. Conclusion

In view of the above amendment, applicant believes the pending application is in condition for allowance.

Dated: May 8, 2006

Respectfully submitted,

By _____

Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
28 State Street
Boston, Massachusetts 02109
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant